Problem A. Raid

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	768 mebibytes

For many year Bitotia was invading Byteotia and robbing it from its natural goods and intelectual properties. However, this time it is Byteotia that will invade sneaky nation of Bitotia. The first step of this carefully planned invasion will be raid on Bitobytian's beach.

Invasion has to be inconspicuous, so k members of elite Byteforce army will be ordered to go on this beach. Currently Byteforce consists of n soldiers numbered with consecutive numbers from 1 to n. Soldier with number i has skill level i in melee combat and skill level a_i in distance combat. Sequence $a_1, \ldots, a_n$ forms a permutation of numbers from 1 to n. The higher skill level, the better soldier is in that particular area.

As everyone knows, in a well formed troop everyone should know who it can give orders to and who it should listen to. If among soldiers that are chosen to participate in invason there are two of them with indices i and j such that $i < j$ and $a_i > a_j$ then it is likely that they will argue about which one is more important. We will call them a *bad pair*.

We would like to omit such arguments and therefore would like to minimize number of bad pairs of soldiers. You should determine what is the lowest number of bad pairs that we can get when choosing exactly k out of n soldiers. Moreover you should determine what is the number of ways in which we can get that lowest number of bad pairs.

And one more thing. It is still not decided how many soldiers should be sent to the Bitobytian's beach. You need to determine these two described numbers for every k from 1 to n.

Input

In the first line of the input there is one integer n ($1 \leq n \leq 40$), denoting the number of soldiers in Byteforce.

In the second line of the input there are n integers $a_1, \ldots, a_n$ ($1 \leq a_i \leq n$, $a_i \neq a_j$ for $i \neq j$), where a_i describes skill level in distance combat of i-th soldier.

Output

You should print n lines consisting of two numbers each.

Numbers in k-th line should denote minimum number of pairs of soldiers that can argue if we choose to send k soldiers to the beach and number of ways in which we can achieve that.

Example

standard input	standard output
5 5 3 1 4 2	0 5 0 3 1 2 3 1 7 1

Explanation to sample test:

If we want to send one soldier only, there will be no arguments obviously and we can do this in five ways.

If we want to send two soldiers then we need to choose one of pairs $(2, 4)$, $(3, 4)$ or $(3, 5)$ in order to have no arguments.

If we want to send three soldiers then minimum number of bad pairs is one and we can achieve that by choosing triples of soldiers $(2, 3, 4)$ or $(3, 4, 5)$.

If we want to send four soldiers then we should choose all but the first one which is the worst in meele combat, but which is the best in the distance combat (since $a_1 = 5$), so he can cause arguments with any other soldier.

If we want to send all five soldiers there will be seven bad pairs.

Problem B. Broken line

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Basia has a string s, each character being one of the first 16 lowercase letters of the English alphabet.

Each character of this string will be replaced by an arrow to the right or up, but the same letters have to be replaced by the same arrow. For example, the string "banan" can be replaced to ↑↑→↑→ or ↑↑↑↑↑, but you cannot obtain →→→↑→ because it would require replacing two letters 'a' with different arrows.

Basia will use the resulting sequence of arrows to draw a broken line. She will start with a pencil set at point $(0, 0)$, then n times she will move it 1 unit right or up – in the direction of the next arrow.

As a *result* of this drawing we will denote the area between the broken line and the OX axis. Formally, this area is a set of points (x, y) such that $y \geq 0$ and there is a point (x, y') that belongs to the broken line and $y' \geq y$ occurs.

What is the largest possible result of Basia's drawing?

Input

The first and the only line of the standard input contains one string s ($1 \leq |s| \leq 300\,000$), consisting of lowercase letters of the English alphabet 'a'-'p' (16 possible characters).

Output

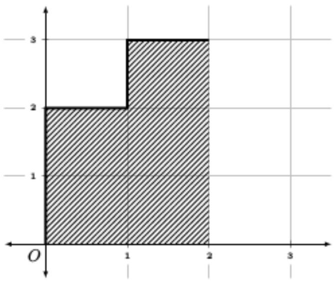
Output one integer – the largest possible result of the drawing obtained after conversion from letters to arrows using given rules.

Examples

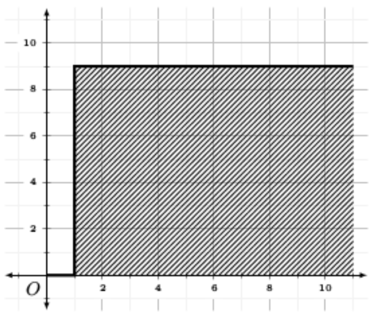
standard input	standard output
banan	5
abcdefghijklmnopaaaa	90

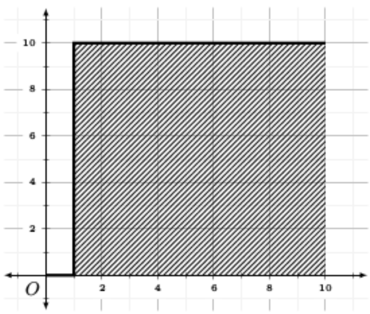
Note:

String "banan" should be replaced with ↑↑→↑→. The area under the broken line is then 5:



For string "abcdefghijklmnopaaaa" there are two optimal solutions with the area 90:





Problem C. Territories

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

This time Byteasar researches a wildlife in a nature reserve that has a shape of an $X \times Y$ rectangle. It is divided into XY unit squares, there is a square with coordinates (x, y) for every $1 \le x \le X$ and $1 \le y \le Y$.

Our hard-working researcher distinguished n species of animals and discovered that each species **dislikes** living on some particular rectangle (which is stricly smaller than whole nature reserve). For species number i it is rectangle described by its two opposite corners (x_i, y_i) and (x'_i, y'_i), for some $x_i \le x'_i$ and $y_i \le y'_i$. We know that there are c_i animals in that species. Therefore, there are $S = c_1 + c_2 + \ldots + c_n$ animals in total.

Byteasar has an idea for a social-natural experiment which relies on putting each of S animals in some cell outside of its disliked region. *Sociality* of a placement is a number of pairs of animals so that both of them are in the same cell. Hence, if a cell contains p animals, this adds $\frac{p(p-1)}{2}$ to the overall sociality.

It is allowed to put animals from the same species into different cells.

Find the maximum value of the sociality that can be attained.

Input

The first line of input contains three integers n, X and Y ($1 \le n \le 100\,000$, $1 \le X, Y \le 1000$) denoting the number of species and dimensions of nature reserve, respectively.

Each of following n lines contains a description of species, i-th of them contains five integers $x_i, y_i, x'_i, y'_i, c_i$ ($1 \le x_i \le x'_i \le X$, $1 \le y_i \le y'_i \le Y$, $1 \le c_i \le 1000$) describing region disliked by species number i and number of animals in that species. For each i at least one of the following conditions holds: $x_i \ne 1$, $y_i \ne 1$, $x'_i \ne X$, $y'_i \ne Y$

Output

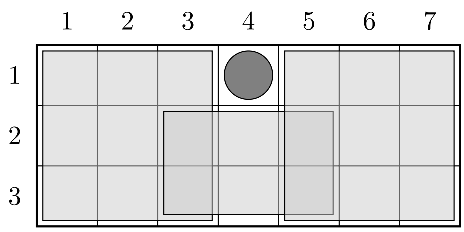
You need to print one integer – the maximum possible sociality of some placement.

Examples

standard input	standard output
2 1 2 1 1 1 1 3 1 2 1 2 4	9
3 7 3 1 1 3 3 1 5 1 7 3 1 3 2 5 3 1	3

Explanation to the sample test: In first sample we need to put four animals in a cell $(1, 1)$ (contributing $\frac{4 \cdot 3}{2} = 6$ to the sociality) and put three remaining animals in a cell $(1, 2)$ (contributing $\frac{3 \cdot 2}{2} = 3$ to the sociality).

Second sample test is depicted below. All animals can be put in a cell $(4, 1)$.



Problem D. Tea

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Bytemommy whole-heartedly loves her Bytekids. However she is kinda forgetful, so instead of giving their proper names, she numbered them with consecutive integers from 1 to n. Everyday she prepares a tea for each of her Bytekids in their favourite cups. One peculiar property of all tea cups in their home is that they have infinite capacity, even though they take finite space only. However, this is for our simplicity only. Bytekid number i prefers to drink exactly l_i bitres of tea everyday. However, amount of tea is not their only requirement — its temperature has to be properly adjusted as well. Bytekid number i would like its tea to have exactly b_i Bytesius degrees.

Unfortunately, one day scatterbrained Bytemommy messed up teas temperatures and temperature of tea in i-th cup was exactly a_i Bytesius degrees, instead of b_i (however i-th kid still got l_i bitres in its cup). Nothing is lost yet — Bytekids are very clever and using some auxiliary cups started to mix up their teas trying to get cups with appropriate amounts and temperatures of teas. You need to determine whether it is possible for Bytekids to reach their goal, that is to get n teas so that i-th tea has exactly l_i bitres and b_i Bytesius degrees.

Formally, Bytekids are allowed to perform following steps arbitrarily many times:

- *Partitioning the tea.* Given a cup with a bitres of tea with temperature t, create two cups of tea with x and $a - x$ bitres of tea with temperature t for some arbitrary real value of x such that $0 < x < a$ (initial cup of tea will no longer exist, obviously).
- *Mixing the tea.* Given two cups of tea with a and b bitres of tea with temperatures t_a and t_b, respectively, create one cup of tea with $a + b$ bitres of tea with temperature

  $$\frac{a \cdot t_a + b \cdot t_b}{a + b},$$

  that is, the weighted mean of initial temperatures (again, initial two cups of tea will no longer exist).

Input

The first line of input contains one integer t ($1 \le t \le 100\,000$) denoting number of testcases.

Description of each testcase starts with a line containing one integer n ($1 \le n \le 100\,000$) denoting number of Bytekids. Following n lines describe Bytekids: i-th of them contains three integers l_i, a_i and b_i ($1 \le l_i, a_i, b_i \le 1\,000\,000$) denoting amount of tea in i-th cup in bitres (both initial and required final one) and initial and required temperature of that tea, respectively.

Sum of values of n over all testcases will not exceed $1\,000\,000$.

Output

You need to print t lines, i-th of them should contain a word `TAK` if it is possible for Bytekids to reach their goal in i-th testcase, or `NIE` otherwise.

Example

standard input	standard output
5 2 2 1 4 2 5 2 2 1 4 3 1 5 4 2 1 5 7 1 7 5 2 1 4 1 1 2 5 3 2 6 4 1 2 3 3 4 5	TAK NIE TAK NIE TAK

Explanation to the sample test: Denote cups of tea as pair of numbers. Pair (l, t) denotes cup with l bitres of tea with temperature t Bytesius degrees.

In the first testcase Bytekids have cups $(2, 1)$ and $(2, 5)$. Using operation of partitioning the tea they can get cups $(\frac{1}{2}, 1)$, $(\frac{3}{2}, 1)$, $(\frac{1}{2}, 5)$ and $(\frac{3}{2}, 5)$. Then, by mixing up cups $(\frac{1}{2}, 1)$ and $(\frac{3}{2}, 5)$, they get $\frac{1}{2} + \frac{3}{2} = 2$ with temperature

$$\frac{\frac{1}{2} \cdot 1 + \frac{3}{2} \cdot 5}{\frac{1}{2} + \frac{3}{2}} = 4,$$

that is — cup $(2, 4)$. Similarly, by mixing cup $(\frac{3}{2}, 1)$ with $(\frac{1}{2}, 5)$, they get $(2, 2)$. In the end, Bytekids will have two cups with appropriate amounts and temperatures of tea.

In the second testcase both teas are too hot. We can't do much here.

However, in the third testcase it is sufficient for Bytekids to swap their cups.

Problem E. Three balls

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	1024 mebibytes

Central European Regional Contest (CERC) is a contest famous for its interesting and always well-prepared tasks. One of these tasks[1] was about finding a volume of a sum of three balls. Maybe it was a challenge 10 years ago, but nowadays contestants should not be bothered with so easy and standard problems. Instead of using 3D space, we will use n-dimensional hypercube. Obviously, it requires some definitions.

n-dimensional hypercube has 2^n vertices, each of them is represented by a sequence of n coordinates which are either 0 or 1. For example, 3-dimensional hypercube has 8 vertices: 000, 001, 010, 011, 100, 101, 110, 111.

Ball with radius r and center s is a subset of vertices of hypercube which have distance at most r to the vertex s. We compute the distance in Manhattan metric which means that vertex p belongs to this ball if and only if coordinates of vertices p and s differ on at most r positions.

Find the number of vertices which belong to the sum of three balls, i.e. number of vertices which belong to at least one of these balls. Print the result modulo $10^9 + 7$.

Input

First line of input contains one integer n ($1 \leq n \leq 10\,000$), denoting number of dimensions.

Description of three balls follow. Each description takes one line and i-th line contains integer r_i ($0 \leq r_i \leq n$) and binary word s_i consisting of n characters which are either 0 or 1. These are the radius and the center of the ball, respectively.

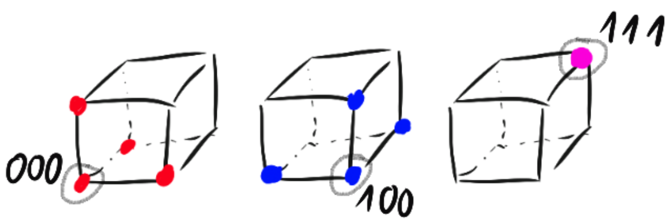
Output

You need to print one integer — number of vertices belonging to sum of these three balls, modulo $10^9 + 7$.

Examples

standard input	standard output
3 1 000 1 100 0 111	7
5 2 10110 0 11010 1 00000	19

Explanation to first sample test: 3-dimensional hypercube is just a mere cube. Following pictures show which vertices belong to following balls. Grey circle denotes center of a ball.



First ball contains vertices 000, 100, 010, 001. Second ball contains vertices 100, 000, 110, 101. Third ball is just a single vertex 111. Sum of these balls contains 7 vertices — all of them except 011.

[1] CERC 2009, problem E: http://cepc09.ii.uni.wroc.pl/lost2.pdf

Problem F. Family photo

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Bitian family is just having a family gathering. Family members are numbered with consecutive numbers from 1 to n. Bityslav is the oldest family member and number 1 was associated to him. Each of the remaining family members has exactly one parent (kinda weird, but let it be).

Such wonderful family gathering needs to be documented well and there needs to be a photo taken. This photo will depict some of family members arranged in a row. However, this year family members are very choosy. Two members agree to stand next to each other if and only if one of them is an ancestor of the second one (not necessarily an immediate one)[2].

Bityslav has a hard task now as he would like as many family members on this photo as possible. What is the biggest number of people that can be present on it?

Input

The first line of input contains one integer n ($2 \leq n \leq 300\,000$) denoting number of Bitian family members. The second line contains $n-1$ integers $p_2, p_3, \ldots, p_n$ ($1 \leq p_i \leq n$, $p_i \neq i$ for $2 \leq i \leq n$), where p_i is an index of a parent of i-th family member. You can assume that there is no cycle in this genealogical tree, i.e. nobody is an ancestor of himself.

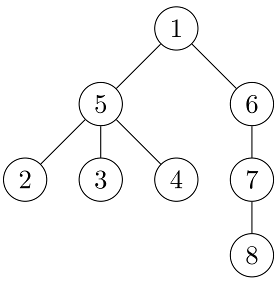
Output

You should output one integer which is the maximum possible number of family members present on a photo with respect to constraints set by family members.

Example

standard input	standard output
8 5 5 5 1 1 6 7	7

Explanation to the sample test: Following picture denotes genealogical tree of Bitian family. One of the optimal photos can contain members with numbers 7, 6, 8, 1, 2, 5, 3 in this order. It can be shown it is impossible to have all of them on one photo.



[2]We say that A is an ancestor of B if and only if A is either a parent of B or a parent of some of B's ancestors

Problem G. Pop music

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	256 mebibytes

Radewoosh loves pop music. It is relaxing, it is great to dance to and even helps in programming. All these advantages, however, require a good tune of the melody with the beat[3].

Radewoosh has just created a new melody and is going to match some good beats to it. The melody lasts n seconds and some of its moments can be much better than others. The quality of the i-th second of the melody is described by the integer a_i (possibly negative). He needs to select the non-negative integers b_i – beat gain coefficients. The coefficient strengthens the second $C(b_i)$-fold, where $C(b_i)$ is the number of *ones* in binary representation of b_i. For example, choosing $b_i = 13$ gives you a threefold gain of i-th second of the melody, because the binary representation of 13 is `1101` and it contains three *ones*.

The final quality of the entire song can be described as:

$$a_1 \cdot C(b_1) + a_2 \cdot C(b_2) + \ldots + a_n \cdot C(b_n).$$

Everyone likes songs with the increasing beat gain and Radewoosh is no exception. The beat gain coefficients must form an increasing sequence of non-negative integers with a certain upper limit of m:

$$0 \leq b_1 < b_2 < \ldots < b_n \leq m.$$

Radewoosh's goal is to choose beat gain coefficients to maximize the final quality of the song.

What is the greatest value he can get?

Input

The first line of the standard input contains two integers n, m ($1 \leq n \leq 200, n - 1 \leq m \leq 10^{18}$) – the length of the song in seconds and the upper limit for the beat gain coefficients.

The second line contains N integers $a_1, \ldots, a_n$ ($-10^{14} \leq a_i \leq 10^{14}$) denoting the quality of the corresponding seconds of the melody.

Output

The output should contain one integer – the maximum possible final quality of the song.

Examples

standard input	standard output
3 5 2 -1 3	9
3 2 1 1 -1	0

Explanation to the first example: The melody consists of three seconds with qualities $2, -1, 3$ respectively. Note that the quality of the second may be negative! The optimal sequence b is $b_1 = 3$, $b_2 = 4$, $b_3 = 5$. Then we get the following quality of the song:

$$a_1 \cdot C(b_1) + a_2 \cdot C(b_2) + a_3 \cdot C(b_3) = 2 \cdot C(3) + (-1) \cdot C(4) + 3 \cdot C(5) = 2 \cdot 2 + (-1) \cdot 1 + 3 \cdot 2 = 9$$

[3] In Polish, "bit" and "beat" are the same word, and then the statement is more entertaining, but that doesn't make that much sense in English. Sorry!

Problem H. Island

Input file:	*standard input*
Output file:	*standard output*
Time limit:	7 seconds
Memory limit:	512 mebibytes

Welcome to island Bitcairn! We have everything here – settlements, roads, beautiful lake, the Internet and a monster living in the lake which is ready to destroy the whole island! Wait, wait, wait – a monster in the lake?!

Byteasar, governor of Bitcairn, has just ordered you to prepare an evacuation plan for tourists in a case of monster's attack. He told you following information about the island:

- The island can be imagined as a ring whose outer boundary is the boundary of a sea and inner boundary is a boundary of the lake.
- There are n settlements numbered with consecutive numbers from 1 to n.
- Settlements with numbers from 1 to a are placed on the boundary of the lake. Moreover they are numbered either clockwise or counterclockwise along that boundary.
- Settlements with numbers from $a + 1$ to $a + b$ are placed on a coast (boundary of sea). Moreover they are numbered either clockwise or counterclockwise along the coast.
- There are m roads connecting settlements. Each of the roads can't pass through neither the lake, the sea nor other settlements. Moreover two roads can intersect in their common endpoint only (if they share one). In other words road network forms a planar graph. Moreover each road can be either directed or undirected.
- All tourists live in settlements that are placed on the boundary of the lake. It can be assumed that from each of these settlements it is possible to get to the coast using road network (possibly using multiple roads).

In order to enable evacuation, you need to design a plan of building seaports. Such plan should contain subset of coastal settlements where seaports will be built and it ensures safety for all tourists if and only if each tourist living in settlement on the boundary of the lake can reach at least settlements on the boundary of the sea where seaports will be built.

Byteasar would like to know, what is the number of plans ensuring safety? Since the result can be huge, it is sufficient to print it modulo $10^9 + 7$. You need to hurry up – safety of tourists depends on you!

Input

The first line of input contains four intergers n, m, a and b ($2 \le n \le 500\,000$, $1 \le m \le 1\,000\,000$, $a, b \ge 1$, $a+b \le n$) denoting number of settlements, number of roads, number of settlements on the boundary of the lake and number of settlements on the boundary of the sea, respectively.

Following m lines describe roads. Each of them contains a description of one road in the following format:

- u_i `--` v_i (describing bidirectional road connecting settlements u_i and v_i)
- u_i `->` v_i (describing unidirectional road going from settlement u_i to settlement v_i)

where $1 \le u_i, v_i \le n$ and $u_i \neq v_i$.

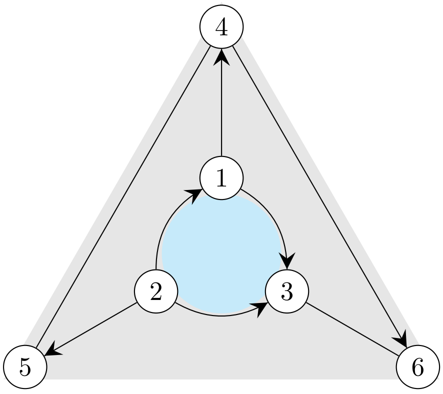
No two roads connect the same pair of settlements. You can assume that these settlements and roads form a planar graph. Moreover for every settlement on the boundary of the lake at least one settlement on the coast can be reached.

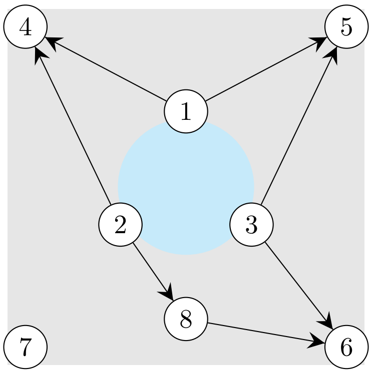
Output

You need to output one integer – number of plans ensuring safety, modulo $10^9 + 7$.

Examples

standard input	standard output
6 8 3 3 2 -> 1 2 -> 3 1 -> 3 3 -- 6 1 -> 4 2 -> 5 4 -> 6 4 -- 5	4
8 7 3 4 1 -> 4 1 -> 5 2 -> 4 2 -> 8 3 -> 6 3 -> 5 8 -> 6	8





Explanation to sample tests: In the first test we need to build a seaport in a settlement number 6 in order to ensure safety for tourists living in settlement number 3. However, tourists from settlements number 1 and 2 are able to reach it as well, so they safety will be ensured as well in that case, so it doesn't matter whether we build seaports in settlements number 4 and 5.

In the second sample test we need to build seaports in at least two out of settlements in numbers 4, 5 and 6 and that is sufficient to ensure safety of all tourists. It doesn't matter whether we build seaport in settlement number 7. It can be readily checked that in total there are 8 plans ensuring safety.

Problem I. Goldfish and pikes

Input file:	*standard input*
Output file:	*standard output*
Time limit:	12 seconds
Memory limit:	256 mebibytes

Byteasar is an owner of a pond with many goldfish. Currently there are n goldfish there, numbered with consecutive numbers from 1 to n. Weight of i-th fish is w_i. Byteasar loves his goldfish and is afraid of evil pikes that could attack his pond (pikes are big fish that are sometimes called "kings of water", similarly as lions are called "kings of jungle").

Byteasar would like to know, to what extent are his goldfish vulnerable to attacks of these predators. He knows biology and psychology of pikes well and that even though they are wild and evil, they are really intelligent as well and that they always attack one at a time. Every pike can be described by two integers — s and k, which denote its current weight of pike and weight that it would like to reach (or even exceed), respectively.

When pike shows up in the pond, he starts to eat goldfish living there. It can eat goldfish if and only if its weight is *strictly bigger* than weight of this goldfish. After eating it, its weight increases by the weight of its victim, which hypothetically can allow it to eat even more goldfish. Mentioned intelligence manifests itself through the fact that pikes always eat minimum number of goldfish that allows them to reach the desired weight.

Byteasar wants to consider many possible scenarios of the attack. Every scenario is a description of a pike that attacks the pond. For each of such scenarios Byteasar would like to know how many goldfish will be eaten by the attacking pike in the case that pike will be able to reach its desired weight. If it won't be able to reach it, it gives up on attacking at all. All these considerations are in fact hypothetical scenarios that are not influencing the state of goldfish in the pond.

Additionally, sometimes new goldfish are brought into the pond. Sometimes they disappear from it as well searching for the happiness in bigger ponds. Byteasar needs to keep all this information updated in order to consider attack scenarios that keep popping into his head. Help him and write a program that will tell him outcomes of all attack scenarios!

Input

The first line of input contains one integer n ($1 \leq n \leq 300\,000$) denoting number of goldfish that live in Byteasar's pond.
The second line of input contains n integers $w_1, \ldots, w_n$ ($1 \leq w_i \leq 10^{12}$) denoting weights of goldfish.
The third line of input contains one integer q ($1 \leq q \leq 100\,000$) denoting number of events that will happen.

Following q lines contain said events. They can come in three types:

- 1 s k – Byteasar considers attack of pike of weight s that wants it to become at least k ($1 \leq s, k \leq 10^{18}$).
- 2 w – One goldfish of weight w is added to the pond ($1 \leq w \leq 10^{12}$).
- 3 w – One goldfish of weight w is removed from the pond. You can assume that there was at least one goldfish of that weight in the pond before that operation.

There is at least one event of the first type.

Output

You need to print one number per every event of the first type which denotes number of eaten goldfish in corresponding scenario, or -1 if pike comes to a conclusion it is not able to reach the desired weight.

Example

standard input	standard output
4 1 4 8 1 15 1 2 3 1 2 4 1 2 5 1 3 3 1 3 5 1 3 16 1 4 16 1 8 17 1 100 101 1 100 115 1 3 9 2 2 1 3 9 3 4 1 3 9	1 2 -1 0 2 4 3 2 1 -1 3 2 -1

Problem J. Tokens

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

We are given a three-dimensional, long and thin board consisting of unit cubes arranged into an $A \times B \times C$ cuboid. Every cell can be described by a triple of integers (i, j, k), where $1 \le i \le A$, $1 \le j \le B$ and $1 \le k \le C$. For every cell we know how many tokens are there initially — in cell (i, j, k) there are $a_{i,j,k}$ of them. In one move we can take one cell that has at least one token and move this token to one of cells $(i+1, j, k)$, $(i, j+1, k)$ or $(i, j, k+1)$, provided that such cell exists.

Moreover, for every cell we are given a number $b_{i,j,k}$. Your task is to determine whether it is possible to perform some number of moves (possibly zero), so that for every cell (i, j, k) number of tokens that end up there is exactly $b_{i,j,k}$.

Input

First line contains an integer t ($1 \le t \le 10\,000$), denoting the number of testcases.

Then descriptions of t testcases follow. Each of them starts with a line containing three integers A, B, C ($1 \le A \le 10\,000$, $1 \le B, C \le 6$), denoting dimensions of the board. Then there are A blocks of B rows. Each of these rows contains C numbers — k-th number in j-th row of i-th block is $a_{i,j,k}$ ($0 \le a_{i,j,k} \le 10^{12}$). Then, in analogous format, numbers $b_{i,j,k}$ are given ($0 \le b_{i,j,k} \le 10^{12}$).

Every testcase contains $2A$ blocks in total. Every two consecutive blocks are separated by an empty line for the sake of readability. Within every testcase, the sum of values $a_{i,j,k}$ is equal to the sum of values $b_{i,j,k}$.

Sum of values of A over all testcases will not exceed $10\,000$.

Output

Output should contain exactly t lines, one per each testcase. k-th line should contain a word `TAK` if in k-th testcase it is possible to find a required sequence of moves from the initial to the final state, or a word `NIE` otherwise.

Example

standard input	standard output
2 2 3 4 2 0 0 1 0 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 4 2 2 2 2 2 2 1 2 1 1 1 1 1 1 2 1 2 2 2	NIE TAK

Explanation to second sample test: Below we present sequence of moves leading from the initial to the final state:

```
2 2         2 2         2 1         2 1         1 1         1 1         1 1
2 1         2 0         2 1         1 1         2 1         1 2         1 2
      ->          ->          ->          ->          ->          ->
2 1         2 1         2 1         2 1         2 1         2 1         1 2
1 1         1 2         1 2         2 2         2 2         2 2         2 2
```

Problem K. Even rain

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

There are n columns of width 1 touching by sides arranged from left to right. Column that is i-th from the left has height h_i.

When the rain falls, water can get stuck in some places. It happens when there are short columns between two high columns.

Formally, water will remain in every point which is neither an interior nor boundary of a column, but for which there exist points on the same height belonging to some columns both on the left and on the right.

We are interested in an area of stuck water. If heights h_i are all integers, this area is an integer as well. Superstition of elderly foretells that happiness is brought when the area of stuck water is divisible by 2.

It seems it will be raining today. However, before that you plan to remove k columns, i.e. change their heights to 0. How many out of $\binom{n}{k}$ possibilities of removing columns will result in area of stuck water being even? Print the result modulo $10^9 + 7$.

Input

First line of input contains two integers n and k ($1 \le n \le 25\,000$, $0 \le k \le \min(25, n-1)$), denoting number of columns and number of columns to be removed.

Second line of input contains n integers $h_1, \ldots, h_n$ ($1 \le h_i \le 10^9$) — heights of columns from left to right.

Output

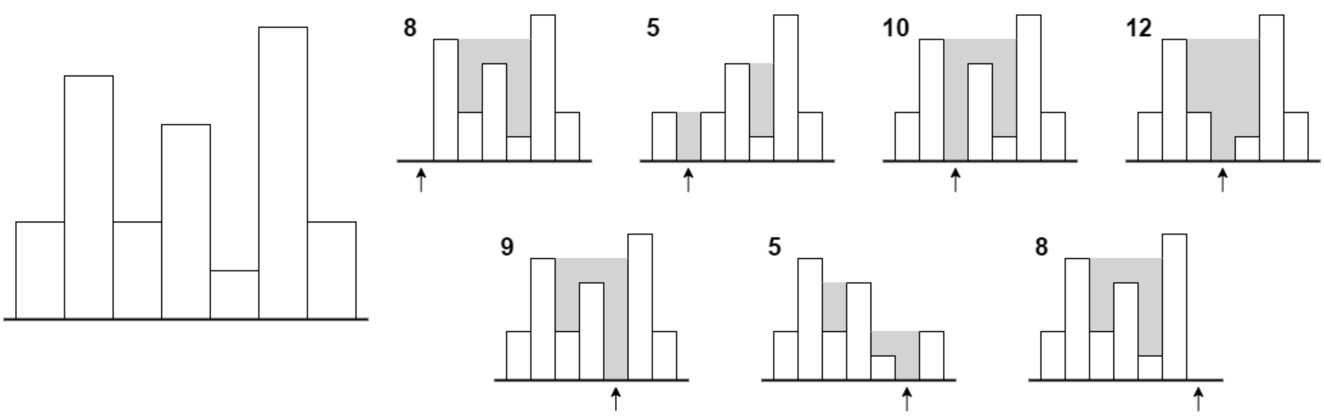
You need to print one number — number of possibilities modulo $10^9 + 7$ to remove k columns so that area of stuck water will be even.

Examples

standard input	standard output
7 1 2 5 2 4 1 6 2	4
5 0 1 3 1 3 1	1

Explanation of first sample test:

Big picture on the left depicts initial setting of columns. Smaller pictures depict each of 7 possibilities of removing one column (pointed by an arrow). Grey areas depict stuck water. In 4 cases area of stuck water (number next to each picture) is even.



Problem L. Floyd-Warshall

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	256 mebibytes

Radewoosh has n-vertex directed weighted graph. He needs to determine distances between all pairs of vertices. He decided to use Floyd-Warshall's algorithm for that.
Correct implementation of Floyd-Warshall's algorithm.

$M - n \times n$ matrix. Initially:

$$M_{i,j} = \begin{cases} 0, & \text{if } i = j \\ w_{i,j}, & \text{if there exists an edge from } i \text{ to } j \text{ with weight } w_{i,j} \\ \infty & \text{otherwise} \end{cases}$$

for $x = 1, 2, 3, \ldots, n$ **do**
  for $y = 1, 2, 3, \ldots, n$ **do**
    for $z = 1, 2, 3, \ldots, n$ **do**
      $M_{y,z} \leftarrow \min(M_{y,z}, M_{y,x} + M_{x,z})$

Unfortunately Radewoosh messed up loops order and his algorithm became incorrect!

Incorrect implementation of Floyd-Warshall's algorithm.
$M - n \times n$ matrix defined as above.
for $y = 1, 2, 3, \ldots, n$ **do**
  for $z = 1, 2, 3, \ldots, n$ **do**
    for $x = 1, 2, 3, \ldots, n$ **do**
      $M_{y,z} \leftarrow \min(M_{y,z}, M_{y,x} + M_{x,z})$

How many distances determined by Radewoosh's algorithm will be incorrect?

Input

The first line of input contains two integers n and m ($2 \le n \le 2\,000$, $1 \le m \le 3\,000$) denoting number of vertices and number of edges in our graph, respectively. Each of the following m lines contains three integers u_i, v_i, w_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$, $1 \le w_i \le 100\,000$) denoting that i-th edge goes from vertex u_i to vertex v_i and has weight w_i. No ordered pair (u_i, v_i) will be given more than once.

Output

Output should contain one number — number of ordered pairs of vertices which have its distance computed incorrectly by Radewoosh's algorithm.

Example

standard input	standard output
4 5 2 3 4 3 4 3 4 2 2 1 3 1 1 2 9	1

Explanation of sample test: Here we depict the following: initial matrix M, matrix generated by correct algorithm and matrix generated by Radewoosh's implementation. Incorrect version made one mistake — $M_{1,2}$.

i\j	**1**	**2**	**3**	**4**
1	0	9	1	∞
2	∞	0	4	∞
3	∞	∞	0	3
4	∞	2	∞	0

i\j	**1**	**2**	**3**	**4**
1	0	6	1	4
2	∞	0	4	7
3	∞	5	0	3
4	∞	2	6	0

i\j	**1**	**2**	**3**	**4**
1	0	9	1	4
2	∞	0	4	7
3	∞	5	0	3
4	∞	2	6	0